Pengelolaan Data Mahasiswa menggunakan NetBeans

Dewi Anisa Hidayanti[1], Yuniana Cahyaningrum [2]

Email:

dewianisaa46@gmail.com[1], yuniana@ikippgribojonegoro.ac.id [2]

Pendidikan Teknologi Informasi, IKIP PGRI Bojonegoro

Abstrak;

Pengelolaan data mahasiswa merupakan tugas penting dalam administrasi perguruan tinggi. Dalam era digital saat ini, penggunaan perangkat lunak yang efisien dan dapat diandalkan sangatlah penting. Salah satu perangkat lunak yang populer dalam pengembangan aplikasi adalah NetBeans. Artikel ini bertujuan untuk memberikan panduan praktis dalam pengelolaan data mahasiswa menggunakan NetBeans. Pengelolaan data mahasiswa merupakan aspek penting dalam lingkungan pendidikan. Data mahasiswa mencakup informasi seperti nama, identitas pribadi, program studi, dan catatan akademik. Pengumpulan, penyimpanan, dan pengelolaan data mahasiswa yang efisien sangat diperlukan untuk memastikan administrasi yang tepat, pengambilan keputusan yang informasional, dan pelaporan yang akurat. Dalam abstrak ini, kami menyoroti pentingnya pengelolaan data mahasiswa dan implikasinya dalam konteks pendidikan. Kami membahas kebutuhan dasar, tantangan yang mungkin muncul, dan solusi teknologi yang dapat digunakan untuk mengelola data mahasiswa secara efektif.

Kata Kunci: Data Mahasiswa, NetBeans, Pengelolaan Data, Aplikasi Pengelolaan, Perguruan Tinggi.

1.Pendahuluan:

Dalam lingkungan perguruan tinggi, terdapat ribuan mahasiswa yang perlu dikelola dengan baik. Informasi-informasi penting seperti nama, nomor induk mahasiswa (NIM), program studi, dan nilai akademik perlu disimpan, diperbarui, dan diakses dengan mudah. Penggunaan NetBeans sebagai platform pengembangan aplikasi dapat membantu mempermudah proses tersebut. Pengelolaan data mahasiswa yang baik memiliki banyak akademik, dan analisis tren pendidikan.

Terakhir, data mahasiswa dapat membantu dalam menyediakan dukungan dan pelayanan akademik yang personal kepada mahasiswa, serta memfasilitasi komunikasi dan kolaborasi antara berbagai departemen dan unit di dalam perguruan tinggi.

Namun, pengelolaan data mahasiswa juga menimbulkan tantangan. Volume data yang besar, kompleksitas informasi, perubahan data yang terjadi sepanjang waktu, dan masalah keamanan data adalah beberapa faktor yang harus diperhatikan. Oleh karena itu, menggunakan teknologi dan sistem informasi yang tepat sangat penting untuk mengelola data mahasiswa dengan efisien dan efektif.

2. Rumusan Masalah

1. **Bagaimana cara menghubungkan NetBeans dengan basis data untuk pengelolaan data mahasiswa?**
2. **Bagaimana cara menyimpan data mahasiswa ke dalam basis data dan mengambilnya kembali saat diperlukan?**
3. **Bagaimana cara mengelola dan menampilkan data mahasiswa secara efisien menggunakan komponen GUI NetBeans seperti tabel atau daftar?**

3.Metode

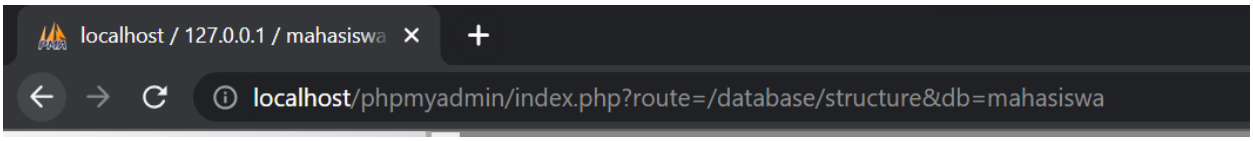
metode yang digunakan adalah menggunakan metode MVC (Model, View dan Controller), MVC merupakan salah satu konsep pemrograman yang membagi aplikasi menjadi tiga bagian, yaitu model yang mengurusi interaksi antara aplikasi dan database, view yang mengurusi urusan logika pemrograman dan controller yang mengatur interaksi antara view dan model.

4.Tinjauan Pustaka

Framework MVC (Model-View-Controller):
Anda dapat menerapkan pola desain Model-View-Controller (MVC) untuk mengelola data mahasiswa. Dalam pendekatan MVC, Anda memisahkan logika bisnis (model), antarmuka pengguna (view), dan logika pengendali (controller). Model akan bertanggung jawab untuk mengelola data mahasiswa dan operasi basis data, sedangkan view akan menampilkan data kepada pengguna, dan controller akan menangani interaksi pengguna dan mengirimkan permintaan ke model.

Apache :
Jika Anda ingin mengimpor atau mengekspor data mahasiswa dari atau ke format file seperti Excel, Anda dapat menggunakan Apache POI. Apache POI adalah pustaka Java yang memungkinkan Anda untuk membaca dan menulis file Microsoft Office, termasuk file Excel. Dengan Apache POI, Anda dapat mengambil data dari basis data dan menulisnya ke file Excel.

5.Hasil & Pembahasan



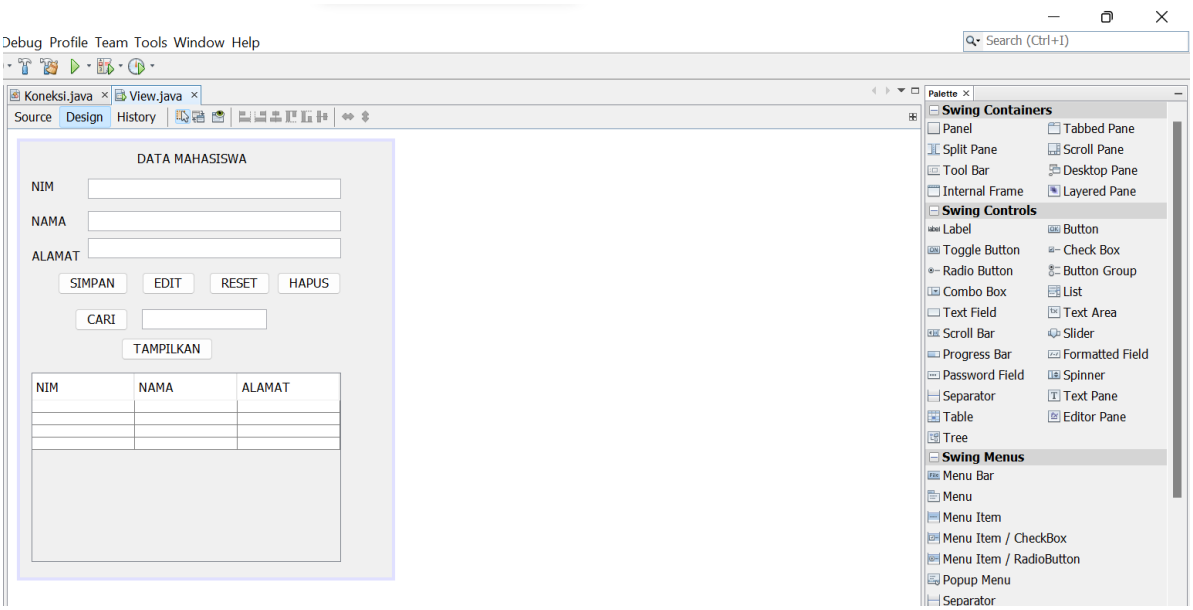
Gambar 1. Tampilan awal membuat data base

Nyalakan mysql dan apache lalu pergi ke google ketik localhost lalu bikin data base mahasiswa

```
29      void cari() {
30           DefaultTableModel tbl = new DefaultTableModel();
31          tbl.addColumn("NIM");
32          tbl.addColumn("NAMA");
33          tbl.addColumn("ALAMAT");
34
35          try {
36              String sql ="SELECT * FROM tb_mahasiswa WHERE nim like '%" +txtcari.getText() + "%'";
37              Connection con = (Connection) Koneksi.getConnection();
38              Statement st = con.createStatement();
39              ResultSet rs = st.executeQuery(sql);
40              while (rs.next()) {
41                  tbl.addRow(new Object[] {
42                      rs.getString("nim"),rs.getString("nama"),rs.getString("alamat"),
43                  });
44                  jTable1.setModel(tbl);
45              }
47          } catch (Exception e) {
47          }
48      }
```

Gambar 2.Atribut yang dimiliki mahasiswa

Tulis atribut mahasiswa yang anda buat di localhost missal NIM,Nama,Alamat dll harus sesuai data yang anda buat biar codingannya ga salah.



Gamabar 3. Moc up/Desain

Kita bikin desainnya dulu untuk tampilan aplikasinya dan desain sesuai data base yang anda buat di localhost.

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package crud;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
/**
 *
 * @author OPERATOR
 */
public class Koneksi {
    private static Connection koneksi;

    public static Connection getConnection() throws SQLException {
        String db ="jdbc:mysql://localhost:3306/data_mahasiswa";
        String user = "root";
        String pass = "";

        if(koneksi == null) {
            koneksi =DriverManager.getConnection(db, user, pass);
        }

        return koneksi;
    }
}

```

Gambar 4. Koneksi

Kita harus bikin koneksi agar tersambung ke data base yang kita buat di localhost.

6.Kesimpulan

Penggunaan konsep MVC akan membuat kode aplikasi menjadi lebih terstruktur. Pembagian atau pemisahan aplikasi menjadi 3 bagian (Model, View dan Controller) membuat programmer lebih mudah melakukan proses pengembangan program karena setiap kode memiliki posisi masing-masing sehingga proses analisa aplikasi menjadi lebih mudah. Pemisahan ini juga memungkinkan aplikasi dikerjakan oleh orang yang berbeda, misal satu orang mengurusi logika dan yang lainya menangani tampilan aplikasi, hal ini dapat terjadi karena dalam konsep ini kode yang berhubungan dengan logika dan kode yang berhubungan dengan tampilan aplikasi terpisah.

DAFTAR PUSTAKA

[1]Basuki. Awan Pribadi, 2011, Membangun Web Berbasis PHP dengan Framewok Codeigniter.
[2] Saputra. Agus, 2012,Trik Dashyat menjadi web master dengan Framework CakePHP. Andi Publisher
Lokomedia.
[3] McCool. Shawn, 2012, “Laravel Starter The definitive introduction to the Laravel PHP web
development framework”. Packtpub
[4] Rosmala. Dewi, Ichwan. Muhammad , Gandlisha. M Irzan, (2013) “Komparasi Framework MVC
Agustus 2011.
[5] Sakur. Stendy B, 2011,PHP 5 Pemrograman Berorientasi Objek Konsep & Implementasi. Andi
Publisher
[6] Wiswakarma Komang, 2010. 9 Langkah Menjadi Master Framework Codeigniter.
Lokomedia